

# Symport/Antiport P Systems with Membrane Separation Characterize $P^{\#P}$

Vivien Ducros    Claudio Zandron

ENS Paris-Saclay, Université Paris Saclay, Gif-sur-Yvette, France

Dipartimento di Informatica, Sistemistica e Comunicazione,  
Università degli Studi di Milano-Bicocca, Milano, Italy

September 3, 2025  
UCNC, Nice

# P systems

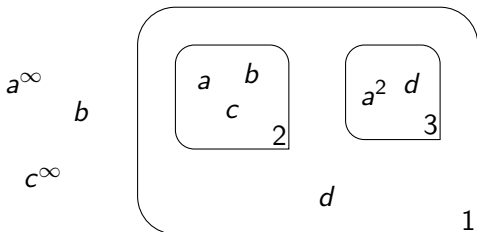
- Model of computation
- Created by G. Păun (2000)
- Bio-inspired
  - ▶ Communication between cells (symport/antiport)
  - ▶ Cell duplication (mitosis/meiosis)
- Many variants!

Here, *Symport/Antiport P Systems with Membrane Separation*.

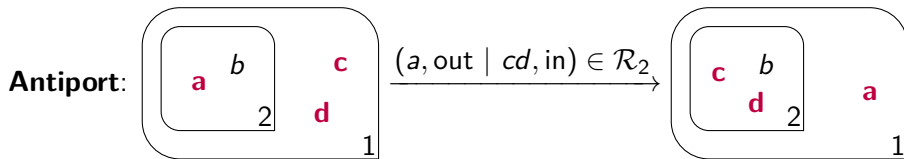
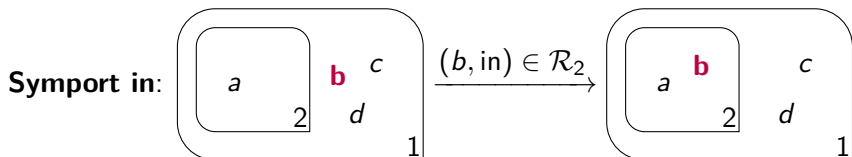
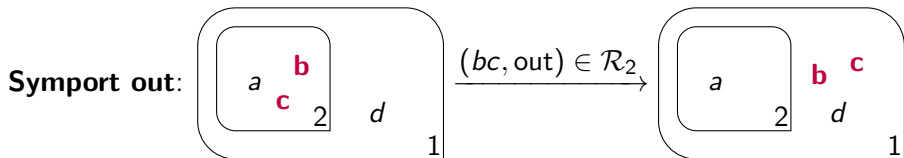
# Definition

A P system is made of:

- Membranes containing membranes... (tree structure)
- Objects (from alphabet  $\Gamma$ )
- Environment objects in arbitrary many occurrences ( $\mathcal{E} \subseteq \Gamma$ )
- Evolution rules: symport, antiport and separation ( $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ )



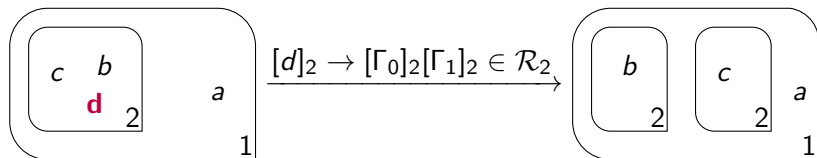
# Communication rules



# Fission rules

## Separation (meiosis):

- partition  $\Gamma = \Gamma_0 \uplus \Gamma_1$ ,
- only for elementary membranes (leaves).



(with  $b \in \Gamma_0$ ,  $c \in \Gamma_1$ )

## Division (mitosis):

- duplication of the content (objects and child membranes),
- all membranes except the skin.

# How does a P system compute?

1. Input : multiset of objects given to the system in membrane  $h_{in}$ .
2. At each time step, apply the rules in the **maximal parallel mode**:
  - the rules apply in parallel,
  - if a rule can be applied (eventually several times), it is applied,
  - objects can be used by only one rule at each step,
  - if a separation rule is applied to a membrane  $h$ , no other rules in  $\mathcal{R}_h$  can be applied.

This is non-deterministic.

3. Computation halts when no rules can be applied.

# P systems solving decision problems

## Definition

A *decision problem*  $X$  is given by: a set  $I_X$  of instances and a function  $\theta_X : I_X \rightarrow \{0, 1\}$ .

An instance  $x \in I_X$  is *accepted* if  $\theta_X(x) = 1$ , else it is *rejected*.

A family  $(\Pi_n)_{n \in \mathbb{N}}$  of P systems solves  $X$  when:

- there exists functions  $s$  (size), and  $cod$  (encoding),
- for  $x \in I_X$ ,  $s(x) \in \mathbb{N}$  and  $cod(x)$  is a multiset of objects,
- computation of  $\Pi_{s(x)}$  on input  $cod(x)$  releases in the **environment** at the **last step** a special object:
  - ▶ **yes**, if  $\theta_X(x) = 1$ ,
  - ▶ **no**, if  $\theta_X(x) = 0$ .

# Complexity classes

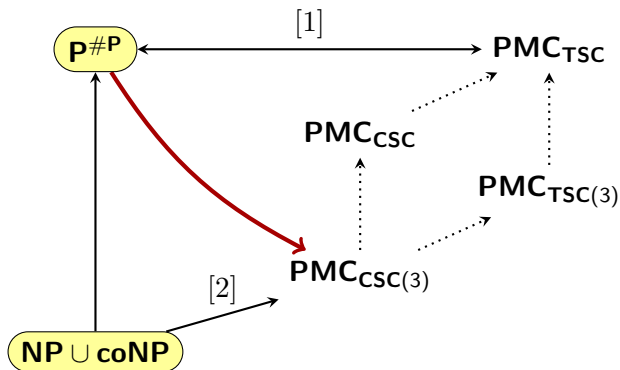
## Definition

**PMC<sub>CSC</sub>**: class of decision problems solved by families  $(\Pi_n)_n$  such that

- $s$  and  $cod$  are computable in polynomial time,
- there exists a Turing machine computing the alphabet, membrane structure, set of rules,... of any  $\Pi_n$  in poly time of  $n$ ,
- $\forall x \in I_X, \Pi_{s(x)}$  halts in a polynomial number of steps on input  $cod(x)$ .

Moreover, if the rules of the P systems have length (= number of objects used by the rule) less than  $k$ , the problem is in **PMC<sub>CSC(k)</sub>**.

# State of the art



**TSC:** tissue-like P systems, where the membrane structure is a *graph*

[1] Leporati, Manzoni, Mauri, Porreca, Zandron (2017)

[2] Valencia-Cabrera, Song, Macías-Ramos, Pan, Riscos-Núñez, Pérez-Jiménez (2015)

# Showing $P^{\#P} \subseteq PMC_{CSC(3)}$

## Definition

The **MIDSAT** decision problem is defined as follows:

**Instance:** A satisfiable boolean formula  $\varphi(x_1, \dots, x_n)$  in CNF.

**Question:** Does the least significant bit of the middle (according to lexicographic order) satisfying assignment of  $\varphi$  equal 1?

## Example

With  $\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ ,  
the satisfying assignments are  $(0, 0, 0); (0, 1, 1); (1, 0, 1); (1, 1, 0); (1, 1, 1)$   
(ordered in the lexicographic order).

Then the median is  $(1, 0, \mathbf{1})$  and the answer to the problem is **yes**.

**MIDSAT** is  $P^{\#P}$ -complete [Toda (1994)]

**Theorem [Ducros, Zandron (2025)]**

**MIDSAT**  $\in PMC_{CSC(3)}$

# Proof of the Theorem

Let  $\varphi$  be a boolean formula,  $n$  number of variables,  $m$  number of clauses.

We design a P system  $\Pi_{n,m}$  solving **MIDSAT**( $\varphi$ ),

→ based on  $\Pi_{n,m}^{\text{SAT}}$  solving **SAT**( $\varphi$ ) by Valencia-Cabrera et Al. (2015)

$\Pi_{n,m}^{\text{SAT}}$  computes in 3 phases:

- **Generation phase:** A membrane separates in  $2^n$  copies encoding all assignments,
- **Checking phase:** Each membrane solves **SAT** independently,
- **Output phase:** The system returns **yes** if  $\varphi$  is satisfiable (else **no**).

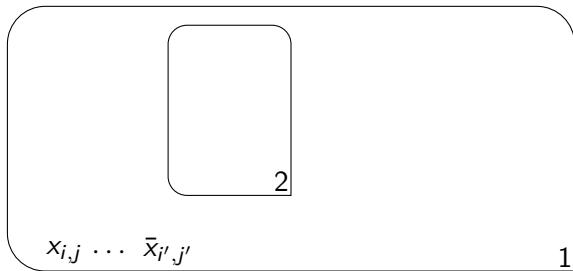
We cut the Output phase, add 2 others (**Activation** and **Search**).

# Solving MIDSAT with P Systems

$$\varphi = \bigwedge_{j=1}^m C_j \text{ with } C_j = l_{1,j} \vee \dots \vee l_{r_j,j} \text{ and } l_{k,j} = x_i \text{ or } \neg x_i$$

**Initial configuration:** two membranes, input multiset contains objects:

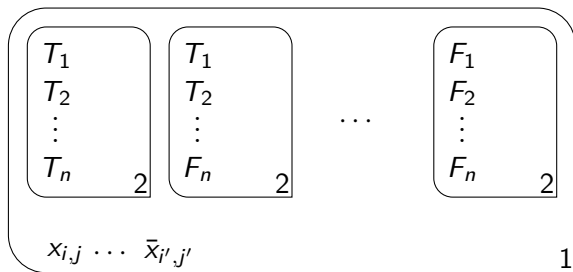
- $x_{i,j}$ , if  $l_{k,j} = x_i$ ,
- $\bar{x}_{i,j}$ , if  $l_{k,j} = \neg x_i$ .



# Solving MIDSAT with P Systems

Generation > Checking > Activation > Search > Output

**Generation phase** ( $3n$  steps): creates  $2^n$  membranes encoding all assignments.

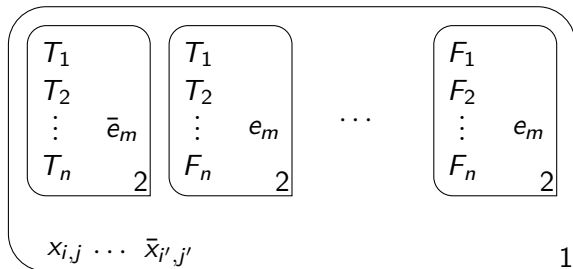


# Solving **MIDSAT** with P Systems

Generation > **Checking** > Activation > Search > Output

**Checking phase** ( $2m$  steps): solves **SAT** in each membrane independently.

- Membranes satisfying  $\varphi$  contain  $e_m$
- Others contain  $\bar{e}_m$

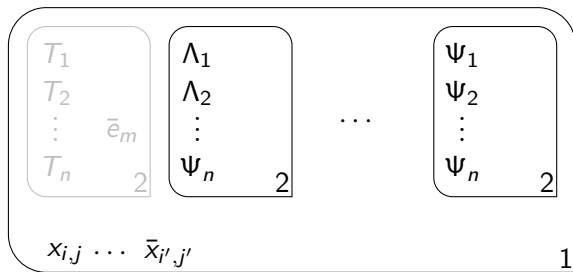


# Solving MIDSAT with P Systems

Generation > Checking > **Activation** > Search > Output

**Activation phase** ( $n + 2$  steps): makes the membranes satisfying the formula able to interact later.

Changes  $T/F_i$  into  $\Lambda/\Psi_i$ .



## Parenthesis: finding the median satisfying assignment

**Similar problem:** finding the median  $b_1 \cdots b_n$  of  $S \subseteq \{0, 1, \dots, 2^n - 1\}$ .  
( $b_1$  most significant bit)

**Key idea** (pseudo-binary search): compute the  $b_k$  progressively.

If we know  $b_1 \cdots b_{k-1}$ , define the **reference assignment**:

$$\nu_k = b_1 \cdots b_{k-1} 10 \cdots 0$$

Then,  $b_k = \begin{cases} 1 & \text{if } \#(S_{\geq \nu_k}) > \#(S_{< \nu_k}), \\ 0 & \text{else.} \end{cases}$

### Example

Satisfying assignments  $S = \{000; 011; \mathbf{101}; 110; 111\}$ .

Suppose we know  $b_1 = \mathbf{1}$  ( $k = 2$ ).

Compare all  $x \in S$  with  $\nu_2 = b_1 10 = \mathbf{110}$ .

$\#(S_{\geq \nu_2}) = 2$  and  $\#(S_{< \nu_2}) = 3$  so  $b_2 = \mathbf{0}$ .

# Solving **MIDSAT** with P Systems

Generation > Checking > Activation > Search > Output

**Search phase** ( $n(3n + 9)$  steps): finds the median among all satisfying assignments, repeats  $n$  times 3 sub-phases.

At  $k$ -th loop,  $b_1 \cdots b_{k-1}$  are encoded in membrane 1 by  $\mu_i / \bar{\mu}_i$ ,  $1 \leq i < k$ .

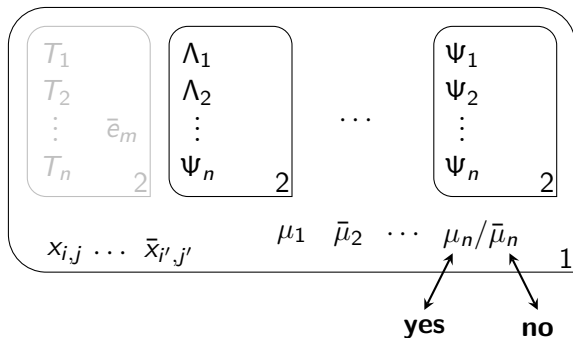
- *Preparation* ( $n + 5$  steps): encodes the reference assignment  $\nu_k$ .
- *Comparison* ( $2n$  steps): compares the satisfying assignments with  $\nu_k$ .
- *Determination* (3 steps): counts the assignments greater/lower than  $\nu_k$  and determines  $b_k$ , encoded as  $\mu_k$  ( $b_k = 1$ ) or  $\bar{\mu}_k$  ( $b_k = 0$ ).

# Solving MIDSAT with P Systems

Generation > Checking > Activation > Search > **Output**

**Output phase** (1 step): returns

- **yes**, if  $\mu_n$  in membrane 1 at step  $3n^2 + 13n + 2m + 2$
- **no**, if  $\bar{\mu}_n$  in membrane 1 at step  $3n^2 + 13n + 2m + 2$



## In details...

- 130+ rules (length  $\leq 3$ ) added to  $\Pi^{\text{SAT}}$   $\rightarrow$  see ArXiv version.
- Implementation in MeCoSim (P system simulator software)  
 $\rightarrow$  on [gitlab.inria.fr/vducros/midsat-csc3](https://gitlab.inria.fr/vducros/midsat-csc3)

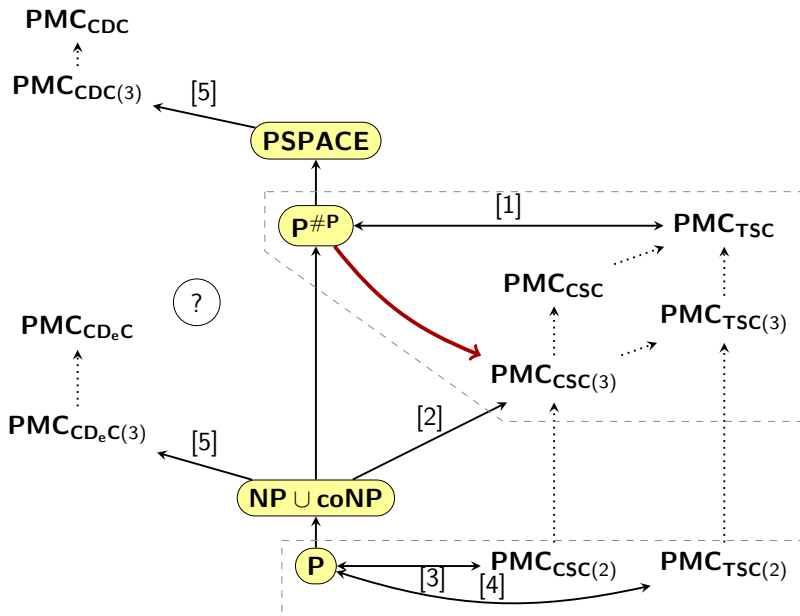
Example of rules added to membrane 2:

$$\mathcal{R}_2'' = \left\{ \begin{array}{l} \{(e_{i,m}, \text{out} \mid \sigma'_0, T_i, \text{in})\}_{i \in \llbracket 1, n \rrbracket} \\ \cup \{(\bar{e}_{i,m}, \text{out} \mid \sigma'_0, F_i, \text{in})\}_{i \in \llbracket 1, n \rrbracket} \\ \cup \{(\sigma'_i, \text{out} \mid \sigma'_{i+1}, \sigma, \text{in})\}_{i \in \llbracket 0, n-2 \rrbracket} \\ \cup \{(\sigma'_{n-1}, \text{out} \mid \sigma, \text{in})\} \\ \cup \{(\sigma, T_i, \text{out} \mid \Lambda_i, \text{in})\}_{i \in \llbracket 1, n \rrbracket} \\ \cup \{(\sigma, F_i, \text{out} \mid \Psi_i, \text{in})\}_{i \in \llbracket 1, n \rrbracket} \\ \cup \{(\Lambda_1, \text{out} \mid p_{k,1}^\wedge, \varphi_k, \text{in})\}_{k \in \llbracket 1, n \rrbracket} \\ \cup \{(\Psi_1, \text{out} \mid l_{k,1}^\psi, \varphi_k, \text{in})\}_{k \in \llbracket 1, n \rrbracket} \\ \cup \{(\Lambda_1, \text{out} \mid \bar{g}_{k,1}^\wedge, \varphi_k, \text{in})\}_{k \in \llbracket 2, n \rrbracket} \\ \cup \{(\Psi_1, \text{out} \mid \bar{p}_{k,1}^\psi, \varphi_k, \text{in})\}_{k \in \llbracket 2, n \rrbracket} \\ \cup \dots \end{array} \right. \begin{array}{l} \text{Rules for the} \\ \text{activation phase.} \\ \\ \text{Rules for the comparison} \\ \text{sub-phase, applied} \\ \text{at step } l_{\text{Comp}}(k) + 1. \end{array}$$

# Implementation in MeCoSim

$n, m$	Formula $\varphi$	Median satisfying assignment	Simulation time (sec.)
3, 3	$(x_1 \vee x_2 \vee \neg x_3)$ $\wedge (x_1 \vee \neg x_2 \vee x_3)$ $\wedge (\neg x_1 \vee x_2 \vee x_3)$	10 <u>1</u>	0.944
5, 4	$(x_1 \vee x_2 \vee \neg x_5)$ $\wedge (\neg x_2 \vee x_3)$ $\wedge (\neg x_1 \vee x_3 \vee x_4)$ $\wedge (x_1 \vee x_4 \vee \neg x_5)$	1001 <u>1</u>	15.458
6, 5	$(x_1 \vee x_2 \vee x_5 \vee x_6)$ $\wedge (\neg x_2 \vee x_3 \vee x_4)$ $\wedge (\neg x_1 \vee \neg x_4 \vee x_6)$ $\wedge (x_1 \vee \neg x_3 \vee \neg x_5)$ $\wedge (x_3 \vee \neg x_6)$	10001 <u>0</u>	53.904
7, 7	$(x_1 \vee x_2 \vee x_3 \vee x_4)$ $\wedge (x_5 \vee x_6 \vee x_7)$ $\wedge (\neg x_1 \vee x_2 \vee x_5)$ $\wedge (x_3 \vee \neg x_5 \vee \neg x_6)$ $\wedge (x_3 \vee \neg x_4 \vee x_7)$ $\wedge (\neg x_1 \vee \neg x_7)$ $\wedge (\neg x_2 \vee x_6)$	010001 <u>1</u>	138.009

# Conclusion



Thank you for listening

Please ask your questions !

## Some references

- [1] Leporati, Manzoni, Mauri, Porreca, Zandron (2017)
- [2] Valencia-Cabrera, Song, Macías-Ramos, Pan, Riscos-Núñez, Pérez-Jiménez (2015)
- [3] Macías-Ramos, Song, Valencia-Cabrera, Pan, Pérez-Jiménez (2015)
- [4] Pérez-Jiménez, Sosík (2015)
- [5] Song, Pérez-Jiménez, Pan (2015)